

[INICIAR SESIÓN](#)[NUESTROS PLANES](#)[TODOS LOS  
CURSOS](#)[FORMACIONES](#)[CURSOS](#)[PARA  
EMPRESAS](#)[ARTÍCULOS DE TECNOLOGÍA > DEVOPS](#)

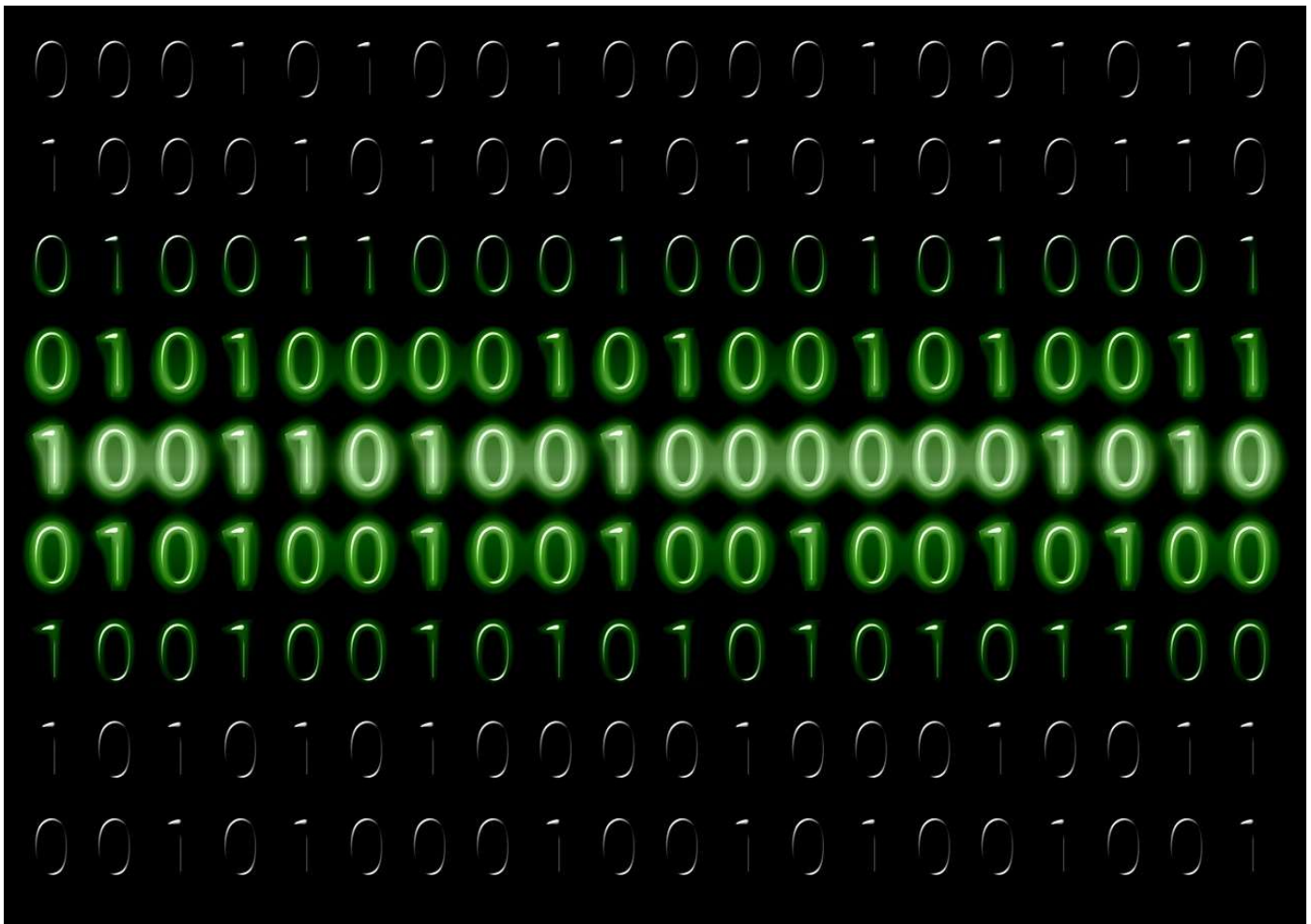
# Como calcular máscaras de subred de forma simple



Yuri-Matheus

21/10/2021

Configurando VLans para segmentar nuestra red, podemos tener varias redes lógicas, con direcciones IPs diferentes.



Sabiendo de eso, cuando fui contratado por una empresa para arreglar la red en un nuevo edificio, me encontré con el siguiente escenario: En el edificio funcionan cuatro departamentos: Financiero, Comercial, RH y Jurídica y cada departamento tienen, en media, 50 hosts.

El departamento comercial envía muchas propuestas técnicas que son archivos bien pesados, por eso la red de todos los departamentos está quedando lenta e inestable.

Además de eso, como todos los departamentos están en la misma red, o sea, en caso de que algún departamento tenga un problema, todos los departamentos serán afectados.

Entonces, vamos atribuir las direcciones de cada departamento. Nosotros podemos utilizar cualquier [IP privado](#) para eso, como son pocos **hosts**. Voy a utilizar las [direcciones de clase C](#) (192.168.x.x).

Podemos atribuir por ejemplo un departamento con la dirección de red 192.168.0.0, otro con 192.168.1.0 y así en adelante.

Esa es una opción totalmente correcta, pero si paramos para analizar, con ese tipo de configuración, vamos a estar desperdiciando direcciones IPs.

## ¿Cómo podemos disminuir ese número de direcciones IPs en la red para evitar ese desperdicio?

Nosotros podemos dividir nuestra red en partes menores, cada una representando un departamento, o sea, dividimos nuestra red en [subredes](#).

## ¿Qué es una subred?

Toda dirección IP tiene una **máscara** correspondiente. Esa máscara que identifica cuál parte de la dirección pertenece a la red y cuál parte al host.

Por ejemplo, la dirección 192.168.0.50 con una máscara 255.255.255.0, identifica como parte de la red la dirección 192.168.0 y el host como 50:

Red: 192,168,0

Host: 50

Para dividir esa red, nosotros podemos “dividir” esa máscara. Para eso, podemos usar valores diferentes entre 0 o 255. Cuando hacemos eso, damos el nombre de [subred](#)

Pero ¿cuáles valores podemos colocar?

## Calculando las máscaras de una subred

Cada tres dígitos en la máscara, corresponden a una parte de la dirección IP de host. Esas partes son formadas por ocho bits, por eso reciben el nombre de octetos.

Cada bit en el octeto, posee un valor en decimal correspondiente a su posición

Por ser bits, solo poseen dos estados, el 0, representando el host, el 1, representando la red.

En este caso, la máscara (255.255.255.0), puede ser representada en binario como:  
11111111.11111111.11111111.00000000.

Máscara: 255,255,255,0

Binario: 11111111,11111111,11111111,00000000

Caso todos los ocho bits de la máscara de subred posean valor 1, el valor en decimal es 255, caso todos los bits fueran 0, posee el valor 0 en decimal.

Pero ¿por qué cuando todos los bits son 1, el valor en decimal que queda es 255?

esa es por causa de la [conversión de binario para decimal](#).

## Creando subred

Vimos que los bits 1 en la máscara especifican la porción de la red y que los bits 0 especifican el host en la dirección IP. Para crear nuestra red, podemos tomar algún **bit 0** y transformarlos en 1.

Por ejemplo, si tomamos el primer bit 0 en la máscara y transformarlo en bit 1, tendríamos la siguiente máscara en binario: 11111111.11111111.11111111.10000000

Y ¿cómo queda el valor de ella en decimal?

Sabemos que los tres primeros octetos tienen el valor en decimal de 255. ya para saber el valor del último, basta realizar la conversión:

$$2^7, 2^6, 2^5, 2^4, 2^3, 2^2, 2^1, 2^0$$

$$1, 0, 0, 0, 0, 0, 0, 0$$

O sea, tenemos:  $1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$ . Que nos deja con 128.

Luego, la máscara de esa red es: 255.255.255.128.

Pero ¿cuántas redes tenemos con esa máscara?

Para saber cuántas redes tenemos con esa nueva máscara basta elevar el número **2**, que son el número de bits posibles, al número de bits prestados de la porción del host.

En este caso tomamos un bit de la porción del host, por tanto, tenemos  $2^1$  redes. O sea, tenemos dos redes.

Y ¿cuántos hosts contiene cada red?

Los hosts son definidos como bits 0 en la máscara de subred, ¿cierto? Entonces para saber cuántos hosts admite nuestra red, basta elevar 2 al número de bits 0 en la máscara.

En este caso tenemos siete bits 0 (10000000), luego,  $2^7$ , que nos da 128.

¿Entonces cada una de nuestras redes componen 128 hosts?

Entre las direcciones IPs, dos necesitan ser reservados. Uno para especificar la red y el otro para especificar el [dominio de broadcast](#). Entonces, en verdad tenemos **128 - 2** hosts, luego tenemos 126 hosts en cada subred.

Y ¿cuáles son esas direcciones reservadas?

La dirección que especifica la red es siempre la primera dirección IP, ya el de **broadcast** es la última dirección IP de la red.

En una red clase C con máscara patrón, la dirección de la red es 192.168.0.0 y el de **broadcast** es 192.168.0.255. Solo que, en nuestro caso, son dos subredes, cada una con una dirección de red y de broadcast.

La primera red posee la dirección de red 192.168.0.0, ya la dirección de broadcast nosotros descubrimos sumando 1 al número de hosts, que en este caso son 126, más la dirección de red. O sea, tenemos **1 + 126 + 0**. De esta forma tenemos que la dirección de broadcast de la primera subred es 192.168.0.127.

ya la dirección de red de la segunda subred es la dirección de **broadcast** de la primera subred más 1. O sea, tenemos como dirección de la red la segunda subred 192.168.0.128. La dirección de broadcast conseguimos obtener de la misma forma.

Sumamos 1 al número de hosts, más la dirección de red. Por tanto, **1 + 126 + 128**, eso nos da la dirección de broadcast 192.168.0.255.

Bueno, calculamos las subredes, pero con esa máscara tenemos apenas dos subredes. Sin embargo, tenemos cuatro departamentos y queremos colocar cada uno en una subred. ¿Cómo podemos hacer eso?

Si nosotros tomamos un bit a más de la máscara de subred del ejemplo anterior. Entonces, tendremos dos elevado a dos, que es el número de bits que tomamos en la porción de host, eso nos deja con cuatro que es el número de subred que queremos.

O sea, nuestra máscara quedará en binario: 11111111.11111111.11111111.11000000.

Realizando la conversión:

$$2^7, 2^6, 2^5, 2^4, 2^3, 2^2, 2^1, 2^0 \quad 1, 1, 0, 0, 0, 0, 0, 0$$

Luego:  $1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$ . Que nos deja con 192.

O sea, nuestra máscara de subred será 255.255.255.192. Pero ¿cuántos hosts admitirá cada subred ?

Para saber eso, basta elevar 2 al número de bits 0 en la máscara, menos los dos bits reservados de cada red, o sea, como tenemos seis bits cero nuestra cuenta queda  $2^6 - 2$ , eso nos deja con 62 hosts para cada red.

Si quisiéramos saber cuál es la dirección de cada subred, basta realizar los mismos pasos de antes.

La dirección de red primera subred será 192.168.0.0, ya su de broadcast será uno más el número de hosts más la dirección de red. eso nos deja con  $1 + 62 + 0$ . Luego, la dirección de broadcast de nuestra primera subred será 192.168.0.63.

De esa forma, después de calcular cada subred tendremos el siguiente resultado:

Red: 192.168.0.0,

Broadcast: 192.168.0.63

Red: 192.168.0.64,

Broadcast: 192.168.0.127

Red: 192.168.0.128,

Broadcast: 192.168.0.191

Red: 192.168.0.192,

Broadcast: 192.168.0.255

Todas esas direcciones utilizando la máscara 255.255.255.192.

De esa forma conseguimos colocar cada departamento en una subred, lo que optimiza la red y aislar los problemas.

## Para saber más

Como cada **subred** es una red distinta, ellas necesitan de un ruteador para que se comuniquen unas con las otras.

En este ejemplo, utilizamos un IP da clase C para crear las subredes, sin embargo, conseguimos realizar esa configuración con todas las clases de IPs disponibles.

Nosotros podemos utilizar otra connotación cuando hablamos de máscaras de rede. En vez de utilizar la máscara en su formato decimal, podemos decir apenas la cantidad de bits 1 contenidos en ella.

Por ejemplo, la máscara patrón de la clase C, 255.255.255.0, en formato binario queda 11111111.11111111.11111111.00000000. O sea, ella tiene 24 bits 1. Entonces en caso que deseemos hablar que la dirección IP 192.168.0.35 utiliza la máscara patrón, podemos utilizar la connotación /24. quedando así: 192.168.0.35 /24.

El nombre de esa connotación es: [CIDR](#).

## Dividir y calcular

Atribuir direcciones IPs a las redes es uno de los trabajos de un **administrador de redes**. Saber cuál dirección atribuir y cuál máscara utilizar hace parte de ese trabajo.

Cuando estamos instalando una red, sería bueno si separásemos los departamentos para aislar posibles problemas, mejorar el desempeño de la red, además de la seguridad.

Una forma de hacer eso es creando **VLans**, por tanto, mismo con esa configuración, algunas redes pueden tener problemas con el broadcast. Entonces una forma de resolver eso es dividiendo la red en partes menores, creando subredes.

Alura Latam tiene variados cursos de [redes](#). En los cuales son presentados los conceptos de subredes y diversas otras situaciones interesantes de utilizar.



Yuri Matheus

Yuri es desarrollador e instructor. Es estudiante de Sistemas de Información en FIAP y graduado como Técnico en Informática en el Senac SP. Su enfoque es en las plataformas Java y Python y en otras áreas como Arquitectura de Software y Machine Learning. Yuri también actúa como editor de contenido en el blog de Alura, donde escribe, principalmente, sobre Redes, Docker, Linux, Java y Python.

ARTÍCULOS DE TECNOLOGÍA > DEVOPS

**En Alura encontrarás variados cursos sobre DevOps. ¡Comienza ahora!**

**SEMESTRAL**

**US\$49,90**

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español



- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

Paga en moneda local en los siguientes países

**ANUAL**

**US\$79,90**

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación

- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

Acceso a todos  
los cursos

Estudia las 24 horas,  
dónde y cuándo quieras

Nuevos cursos  
cada semana

## NAVEGACIÓN

PLANES  
INSTRUCTORES  
BLOG  
POLÍTICA DE PRIVACIDAD  
TÉRMINOS DE USO  
SOBRE NOSOTROS  
PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN  
FRONT END  
DATA SCIENCE  
INNOVACIÓN Y GESTIÓN  
DEVOPS

AOVS Sistemas de Informática S.A  
CNPJ 05.555.382/0001-33

## SÍGUENOS EN NUESTRAS REDES SOCIALES



## ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

## CURSOS

### Cursos de Programación

Lógica de Programación | Java

### Cursos de Front End

HTML y CSS | JavaScript | React

### Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### Cursos de DevOps

Docker | Linux

### Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics | Liderazgo y Gestión de Equipos | Startups y Emprendimiento